

USB - Beyond the Specification

**Brad Hosler
Senior Staff Engineer
Intel Corporation**

Agenda

- ◆ Dealing with bandwidth
- ◆ USB power management
- ◆ USB Specification Rev 1.1
- ◆ USB Implementation Topics
 - ◆ Bus powered hubs
 - ◆ PCI interactions
 - ◆ String descriptors
 - ◆ Extension cables
 - ◆ Current limiters

PnP Evolution

Interconnect	ISA	PNP-ISA	PCI	USB	1394
	Inside the box			Outside the box	
Resources	IO	IO	IO	Bandwidth power	Bandwidth power
	MEM	MEM	MEM		
	IRQ	IRQ	IRQ		
	DMA	DMA	Bandwidth		

Progressively easier for the end user



Requires more and more HW/SW involvement

USB Bandwidth



- ◆ USB frame (1 millisecond) allows up to 1500 bytes of traffic
 - ◆ Isochronous and interrupt traffic have guaranteed bandwidth
 - ◆ Control traffic is “best effort”
 - ◆ Bulk uses what is left

Bandwidth Allocation

- ◆ Lack of bandwidth can lead to situations where a device isn't usable
 - ◆ Bandwidth is allocated on a first-come, first-serve basis
 - ◆ There may be System/OS-specific preset limits for a single device
 - ◆ No automatic “rebalance” when new device is added

Bandwidth Management Features

- ◆ **Devices/drivers MUST provide features to help manage bandwidth**
 - ◆ **Multiple alternate settings for each interface**
 - ◆ **Alternate setting 0 uses no bandwidth**
 - ◆ **Only consume bandwidth when device is being used**

Bandwidth - Needed Features

- ◆ **Driver interface to request a reduction in bandwidth**
 - ◆ **Currently do an unplug/plug cycle**
- ◆ **Graphic application**
 - ◆ **Shows USB bandwidth used**
 - ◆ **Provides adjustment options**
- ◆ **Manage your bandwidth or it will manage you!!!**

Power Management

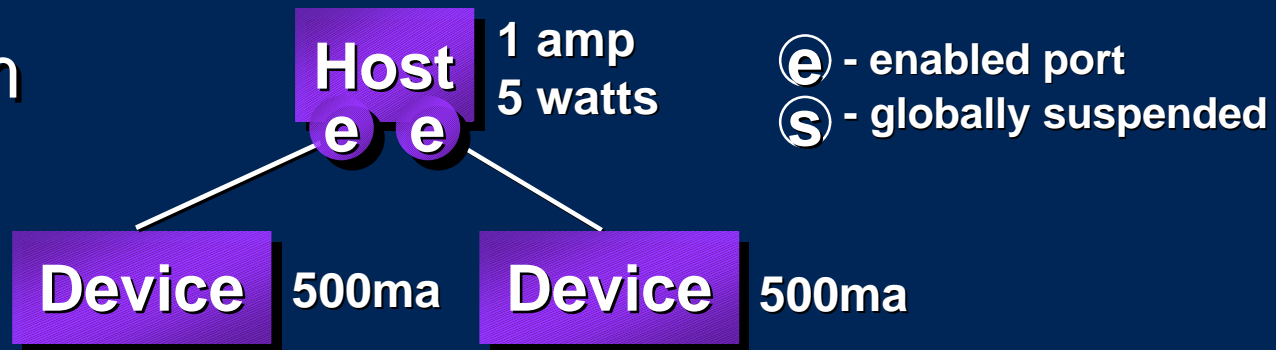
- ◆ Managing power on PCs is receiving lots of attention
 - ◆ Energy Star™ program
 - ◆ Has potential to save users large amounts of money
- ◆ Five Watt PC is the goal
 - ◆ PC appears off but is quickly ready to use
 - ◆ Like a VCR or TV
 - ◆ Can respond to external events (phone, LAN, remote control)

USB Power Management

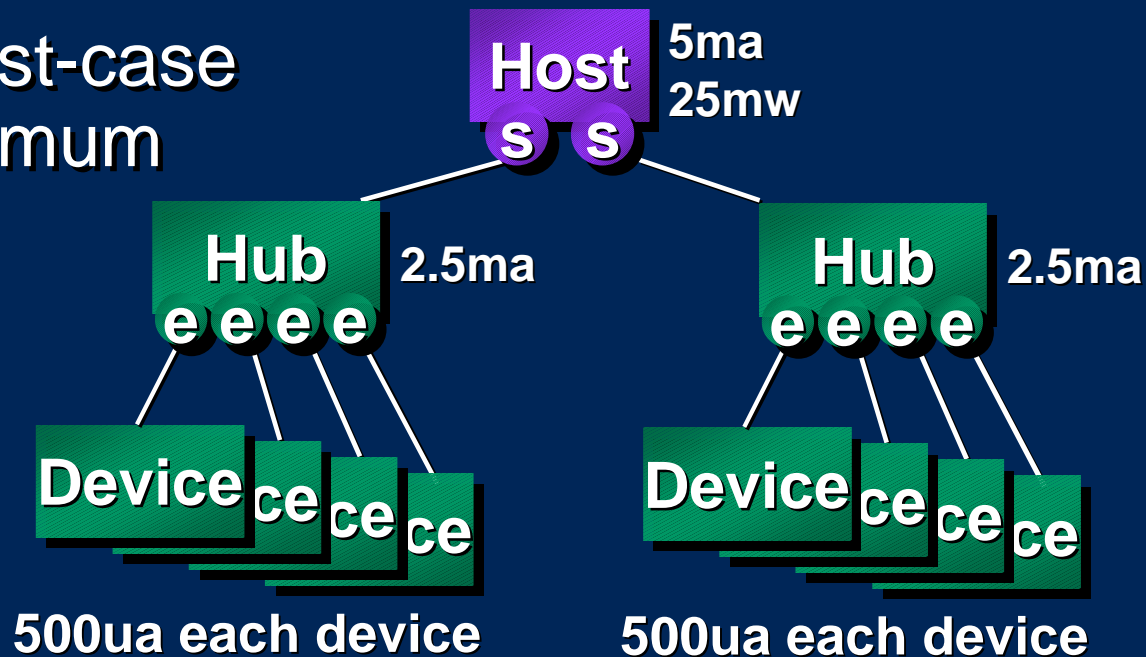
- ◆ Basic USB device requirements
 - ◆ 500mA max current
 - ◆ 100mA max current before being configured
 - ◆ 500uA max when suspended
- ◆ Hubs provide
 - ◆ Selective suspend
 - ◆ Port power control (ganged and individual)

Host Power

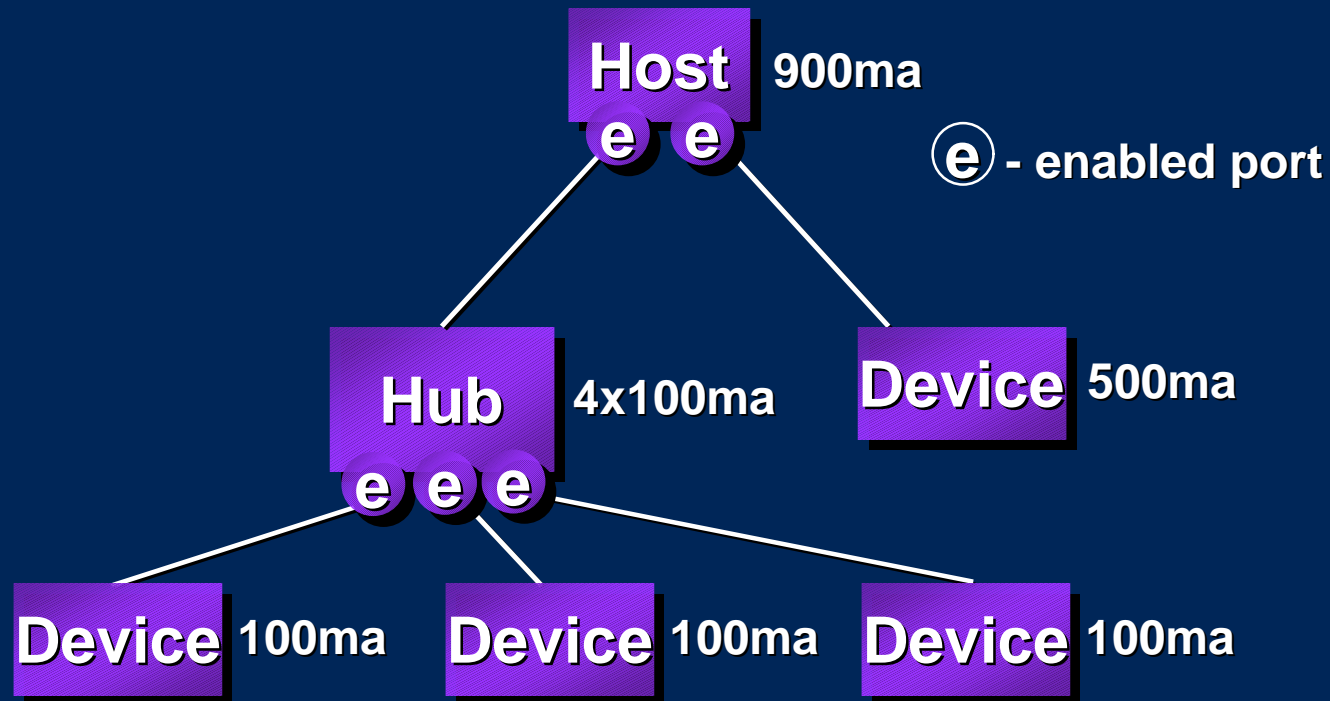
Maximum



Worst-case minimum

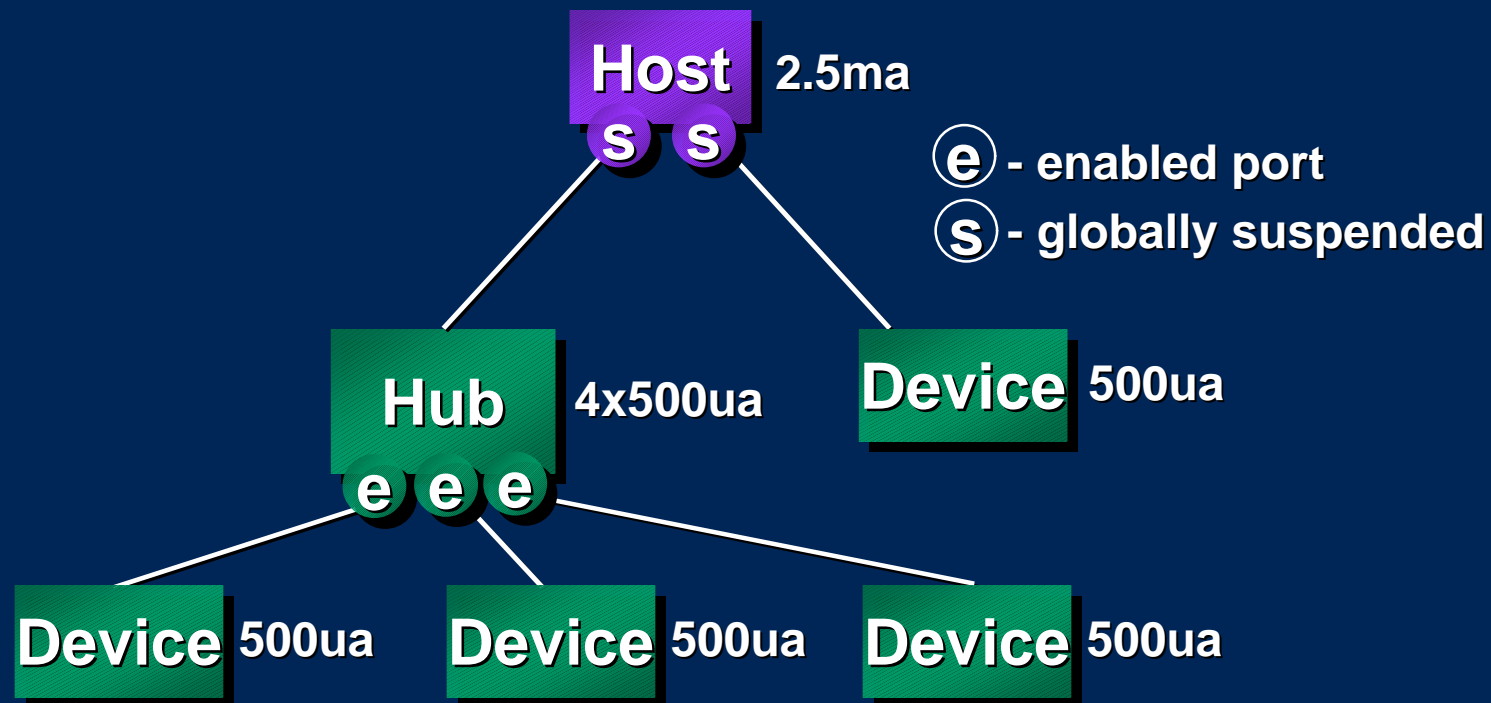


Full Power Example



- ◆ Total current = 900 mA; 4.5 Watts

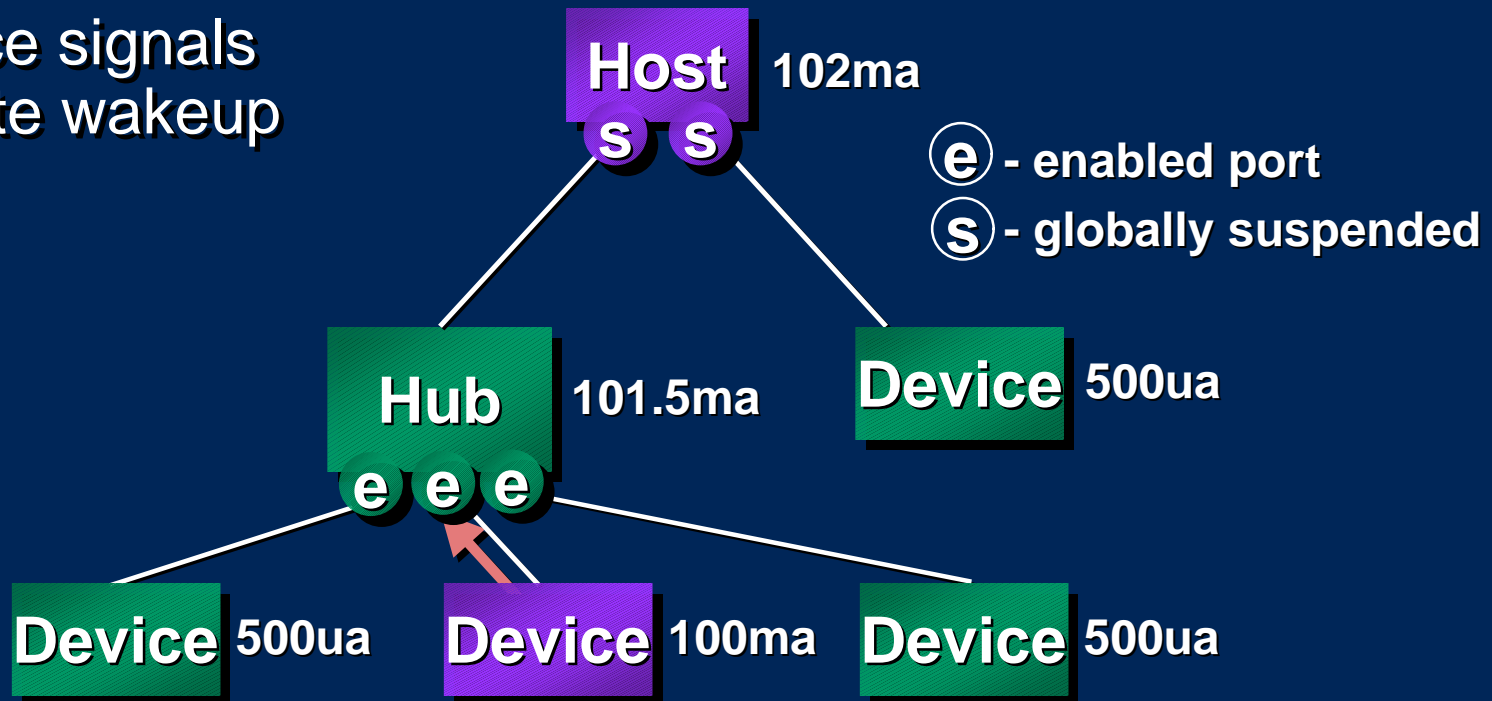
Global Suspend



- ◆ **Total current = 2.5mA; 12.5 mWatts**

Remote Wakeup Example

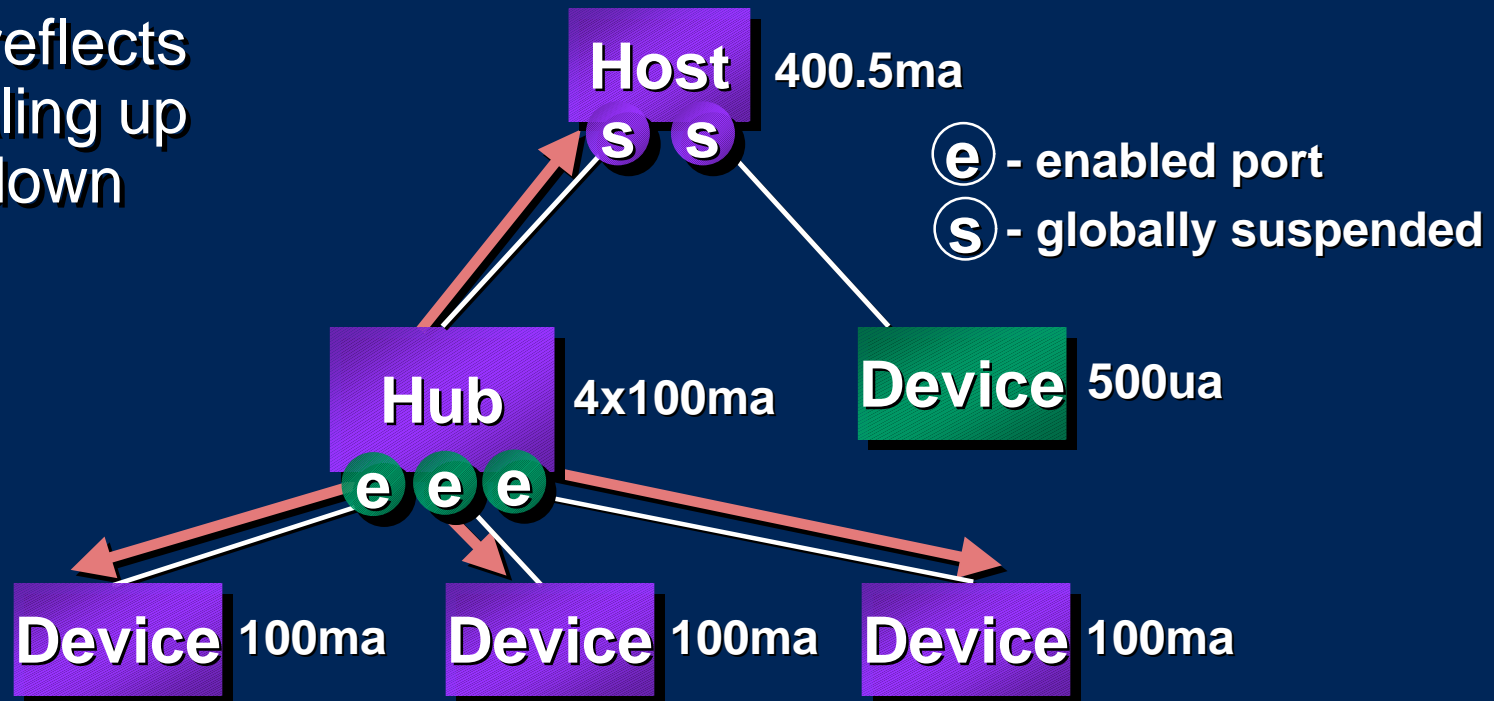
Device signals
remote wakeup



- ◆ Total current = 102mA; 512 mWatts

Remote Wakeup Example

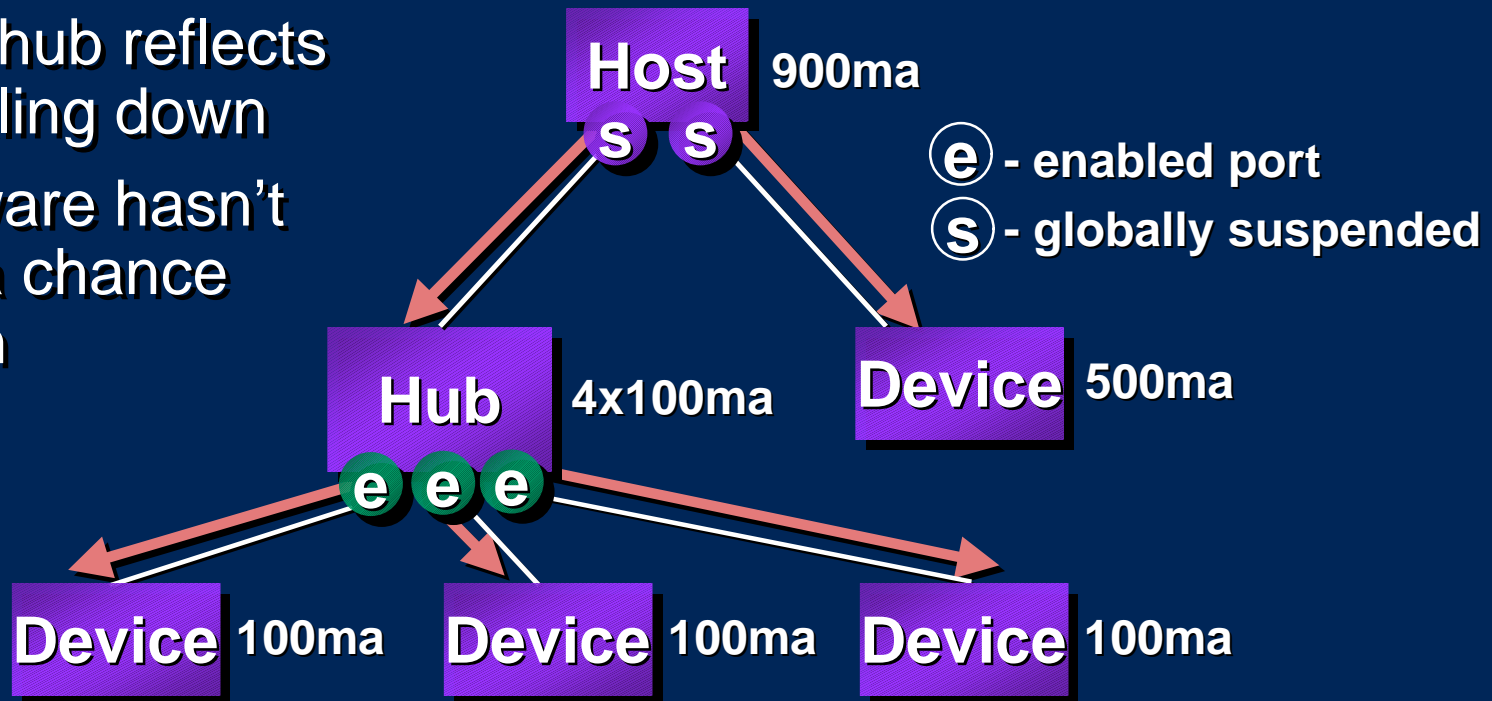
Hub reflects
signaling up
and down



◆ Total current = 400.5mA; 2+ Watts

Remote Wakeup Example

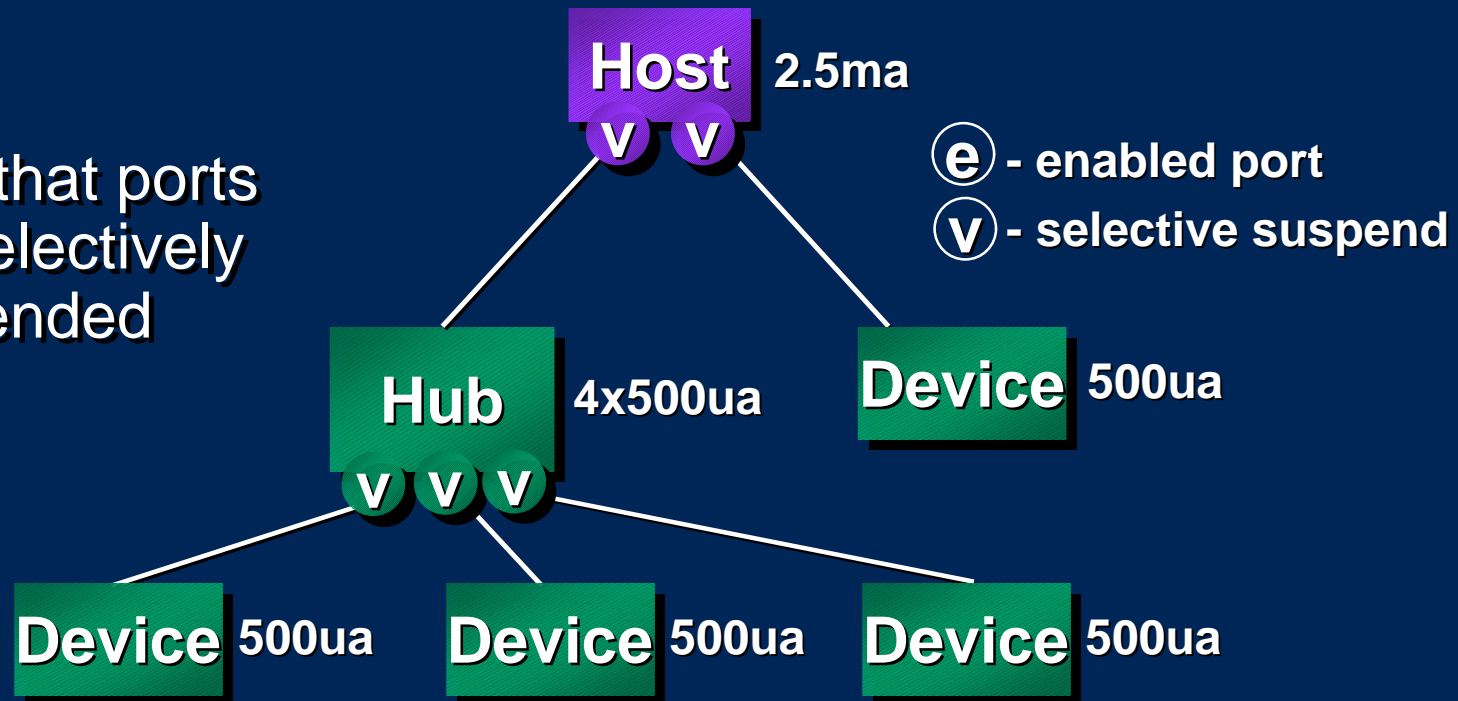
Root hub reflects
signaling down
Software hasn't
had a chance
to run



- ◆ Total current = 900mA; 4.5 Watts
- ← Too large

Selective Suspend Example

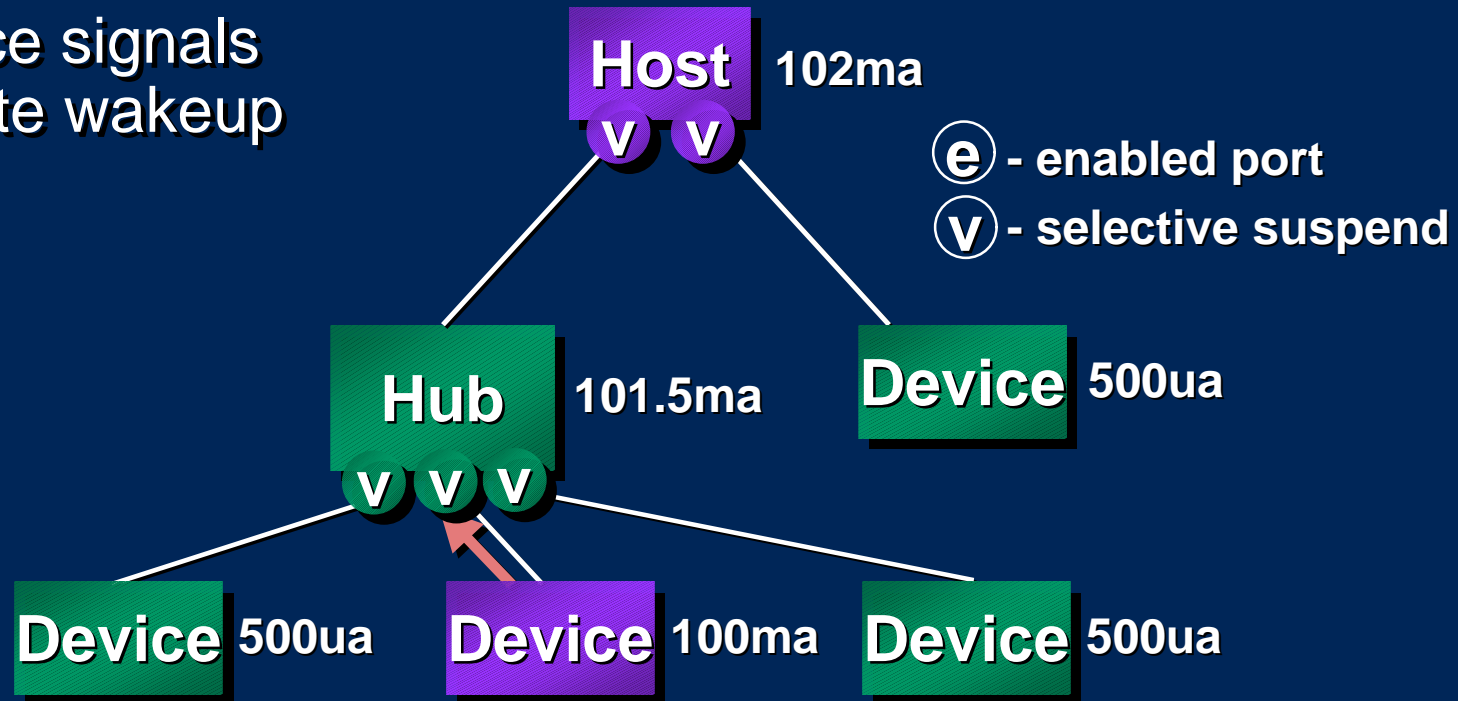
Note that ports are selectively suspended



- ◆ Total current = 2.5mA; 12.5 mWatts

Selective Suspend Example

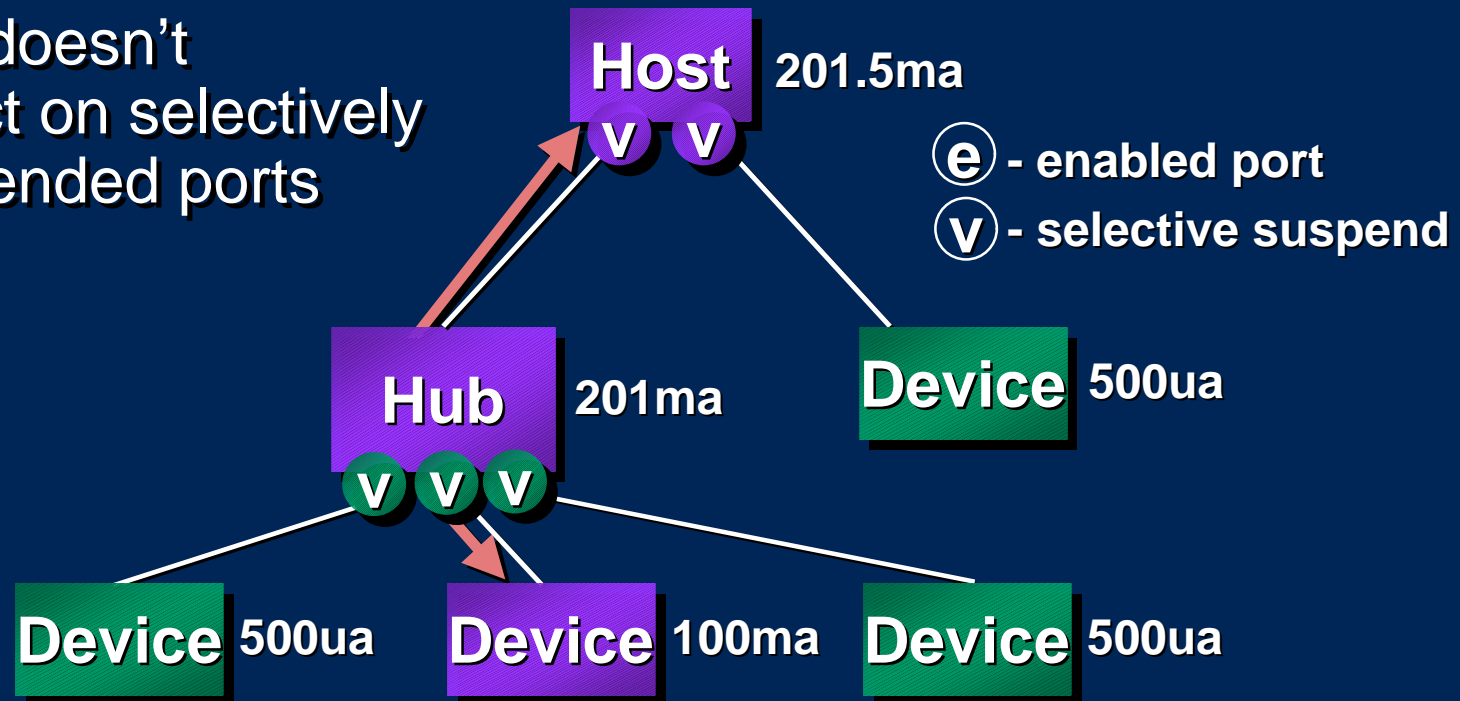
Device signals
remote wakeup



- ◆ Total current = 102mA; 510 mWatts

Selective Suspend Example

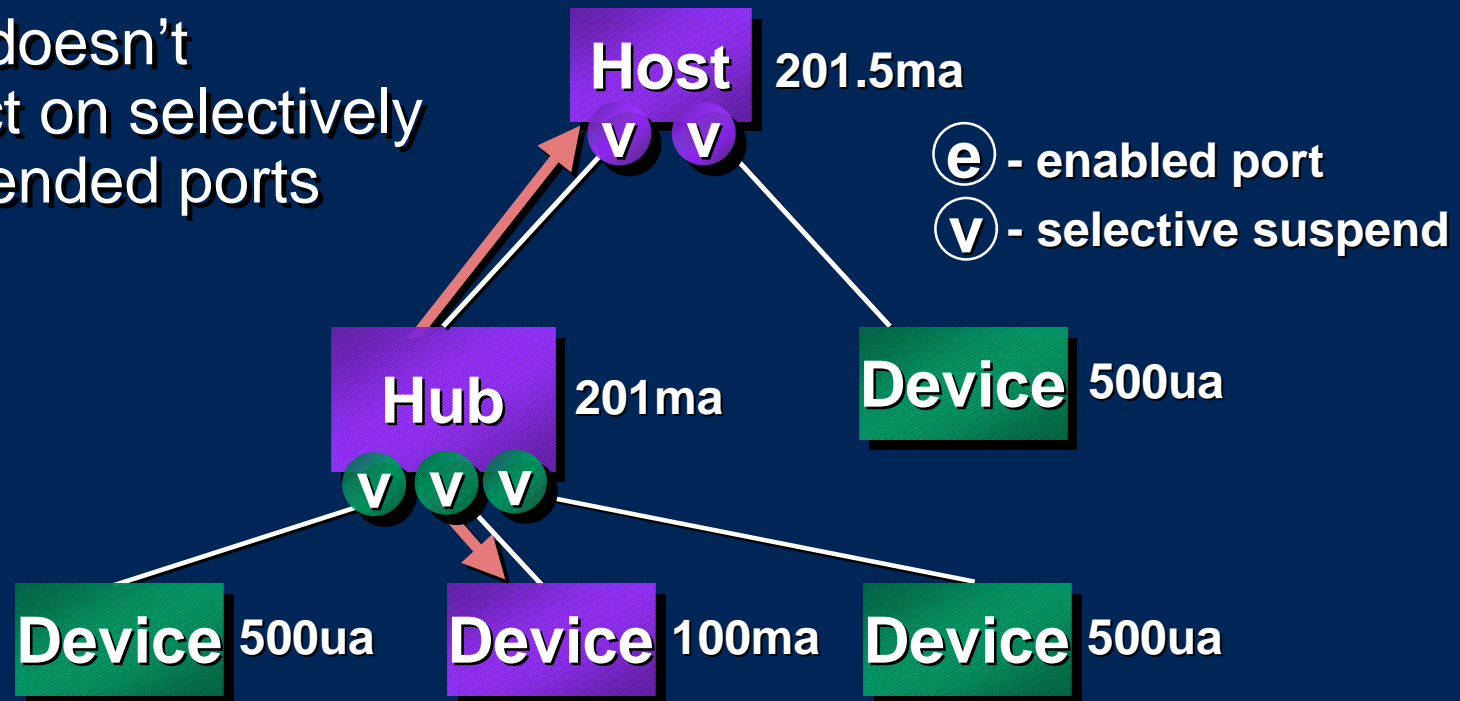
Hub doesn't
reflect on selectively
suspended ports



- ◆ Total current = 201.5mA; ~1.07 Watts

Selective Suspend Example

Hub doesn't
reflect on selectively
suspended ports

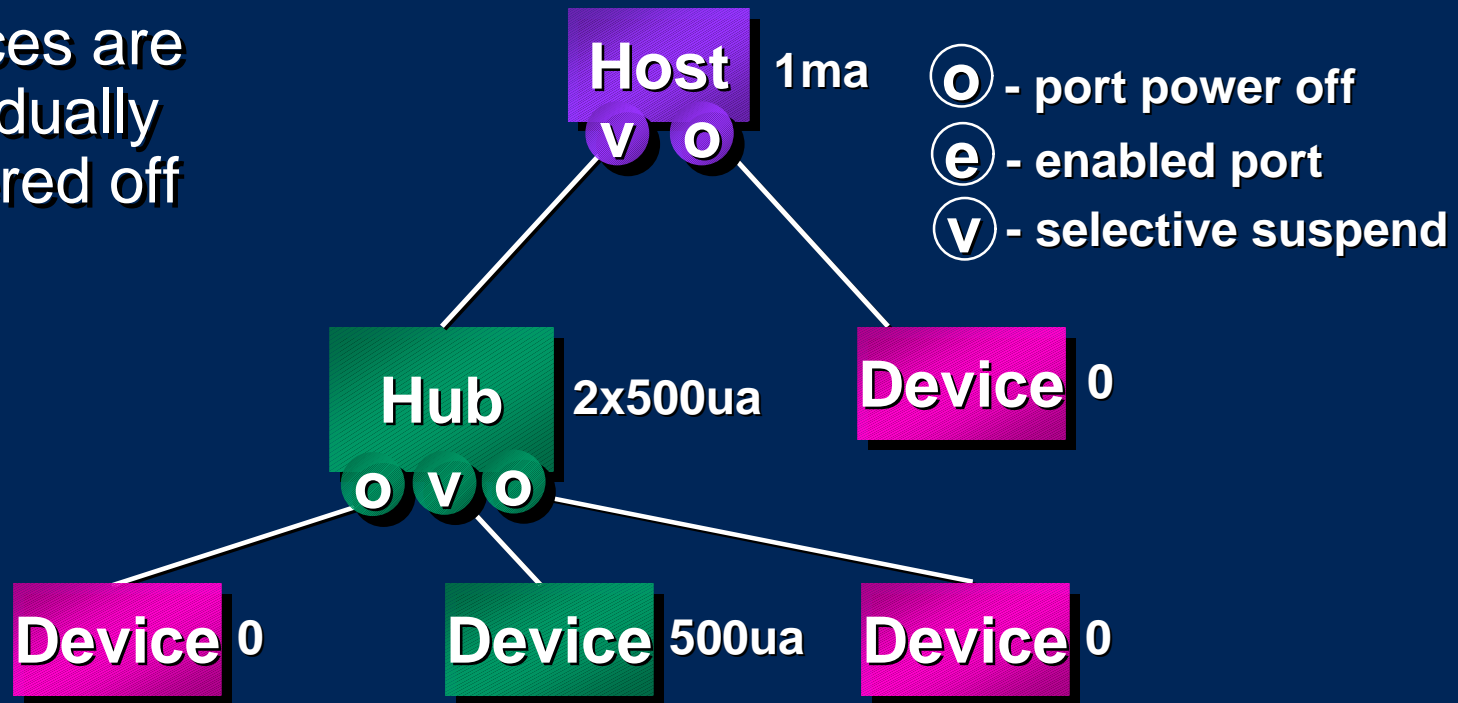


- ◆ Total current =
201.5mA; ~1.07 Watts

← Within
budget

Aggressive Power Mgmt

Devices are individually powered off



◆ Total current =
1 mA; 5mWatts

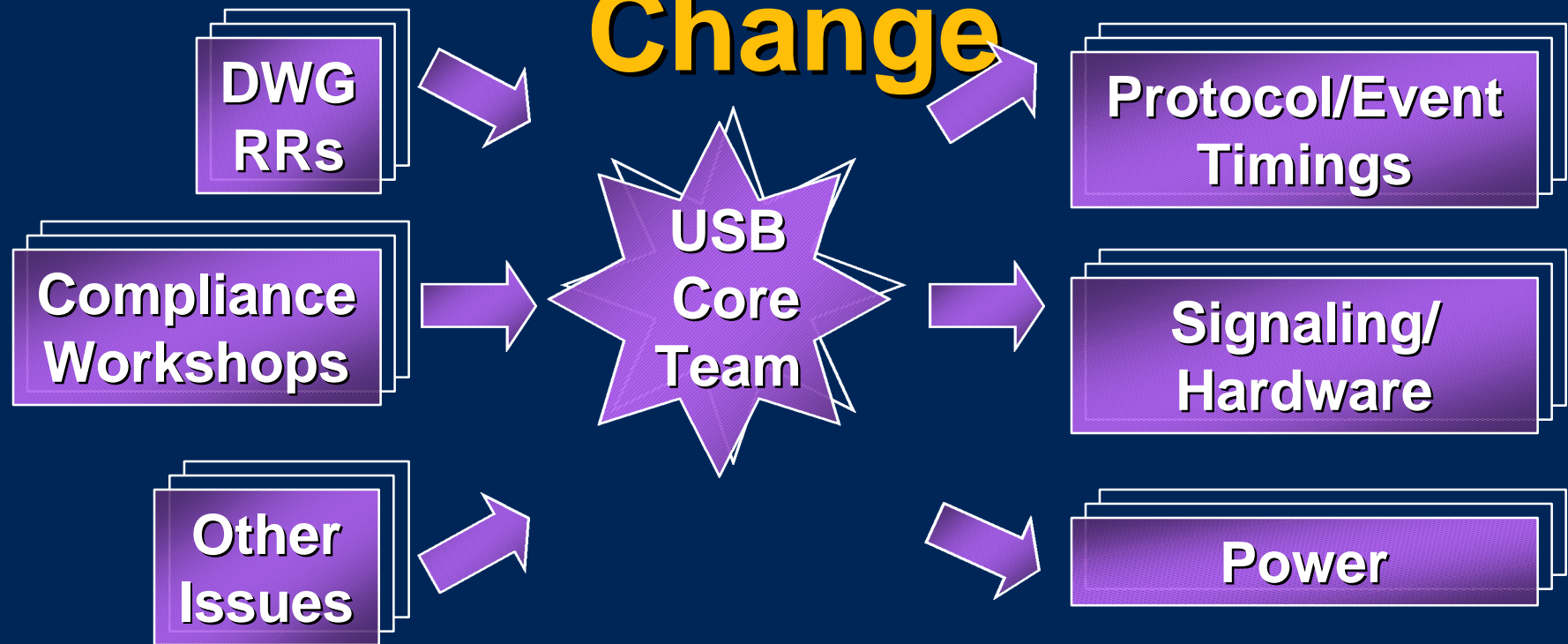
← 60%
better

Device Drivers and Power Management

- ◆ Drivers should be capable of aggressively managing power
 - ◆ Respond to system messages
 - ◆ Automatically limit power when not in use
- ◆ Managing power is important!!!

Universal Serial Bus Specification 1.1 Update

Source Material For Change



Change Classifications:

Clarifications

Enhancements

***Problem
Solutions***

Rules for Applying Changes

- ◆ **Interoperability of 1.0 and 1.1 Devices**
- ◆ **Zero impact on non-hub devices****
- ◆ ***Clarifications:***
 - ◆ Always resolved in favor of 1.0 implementations
- ◆ ***Enhancements:***
 - ◆ simple, natural extensions to USB
- ◆ ***Problem Resolutions:***
 - ◆ 1.0 incorrect or too vague; Technical omissions; Solutions implemented in Hubs

****Goal, Not Guarantee**

1.0 Issues Resolved in 1.1

- ◆ Cables & connectors
- ◆ Remote wakeup
- ◆ Transfer Type Enhancement
- ◆ Power Levels
- ◆ Connection/Power-up Events
- ◆ Electrical Parameters

USB Cables & Connectors

- ◆ 1.0 Chapter 6 is guideline, not spec
 - ◆ Allowed low quality connectors leading to bad user experience
 - ◆ Wasn't explicit about non-standard detachable cables
- ◆ 1.1 Chapter 6 updated to spec language

**Good Specification Ensures
Reliable User Experience**

Remote Wakeup

1.0 Specification

- ◆ Corner cases in USB 1.0
- ◆ Conflicting signal durations
 - ◆ Maximum remote wakeup duration overlaps minimum reset duration
- ◆ By specification:
 - ◆ Remote-wakeup events can get lost
 - ◆ Resume events observable by device are inconsistent

Potential Annoyance for Users

Remote Wakeup

1.1 Specification

- ◆ Fixes all 1.0 issues
- ◆ Impacts only hub implementations

Transfer Type Extension

New for 1.1

- ◆ Interrupt Host to Device (Interrupt OUT) is now legal
- ◆ Free to implement
 - ◆ All host controller implementations allow
 - ◆ Transparent Operating system support
 - ◆ Simple modification to endpoint descriptor

**Easy, simple, and efficient
extension that enables useful,
exciting USB Devices**

Power Levels

New for 1.1

- ◆ **Clarification: Worst-case suspend case is Bus-Powered Hub**
 - ◆ 4 ports @ 500uA + hub controller @ 500uA
- ◆ **New features:**
 - ◆ Suspend is average, not Max
 - ◆ Suspend is 500 uA if not enabled for remote wakeup.
 - ◆ Suspend is 500uA per 100mA of configured power when enabled for remote wakeup

Connection/Powerup Events

- ◆ Responsibilities of Device, Hub and Host software clearly specified
 - ◆ Power on to Power Good time
 - ◆ Maximum time to signal attach
 - ◆ Connection debounce

All Information In One Place

Electrical parameters

- ◆ Relaxation
 - ◆ Differential receiver design
 - ◆ First bit propagation through hub
- ◆ Clarification
 - ◆ Overcurrent detection
 - ◆ *Test conditions*

**Restrictive or implicit requirements
in 1.0 relaxed or clarified in 1.1**

USB 1.1 Summary

- ◆ Incremental update; solving problems
- ◆ Updates based on industry input
- ◆ All 1.0 devices will inter-operate with 1.1
- ◆ 1.0 devices need not change

1.0 is Solid Specification
It's only getting Better!
Don't Wait for 1.1

Bus Powered Hubs

Bus Powered Hubs

- ◆ Introduce topology restrictions
 - ◆ can't support high power devices
 - ◆ can't have two in series
- ◆ Problem indication to user is minimal
- ◆ Software add-ons can dynamically detect problem and guide end user to solution

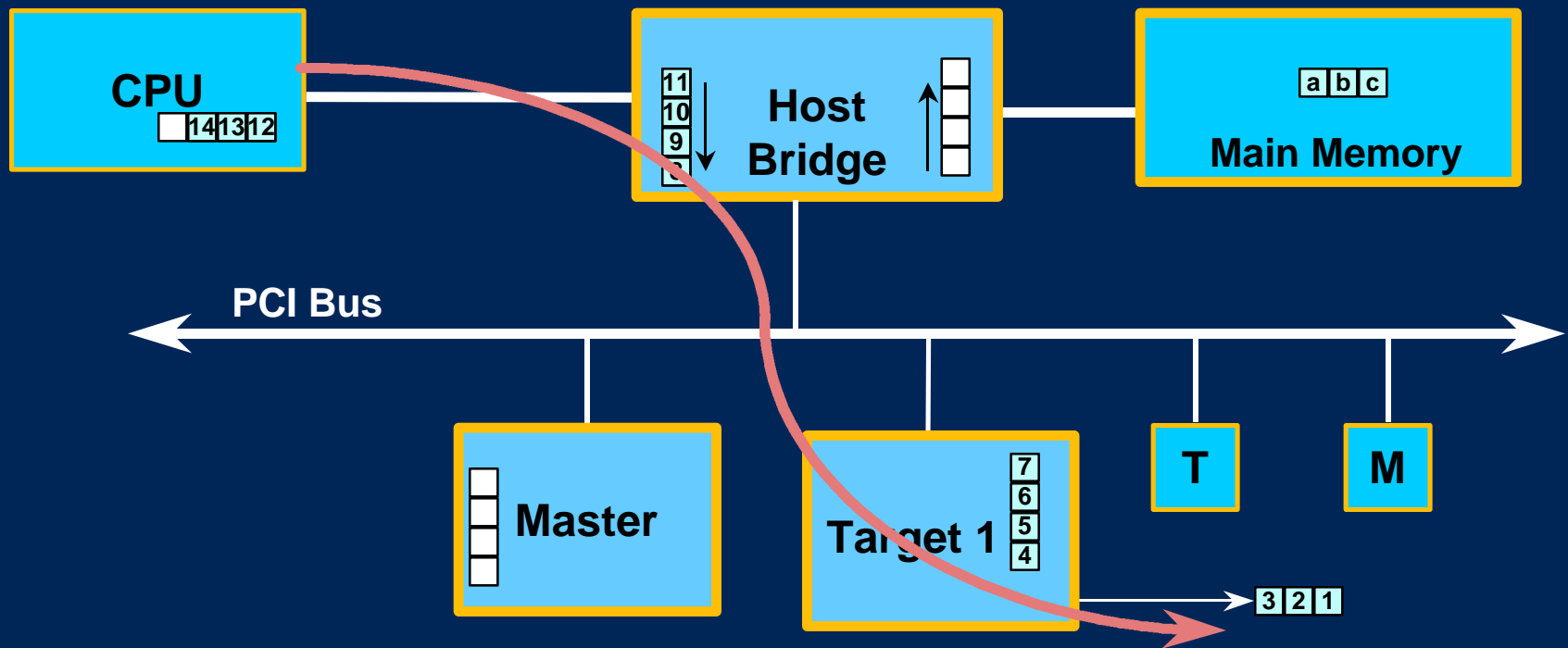
PCI Usage Issue

PCI 'bad citizen' syndrome

- ◆ PCI 'bad citizens' misuse PCI bandwidth
 - ◆ Memory Writes are retried for extended times (up to 12 milliseconds)
- ◆ PCI misuse keeps USB Host Controllers from accessing memory
 - ◆ Can't properly service ISOCH channels
 - ◆ Results in video banding, audio clicks/pops

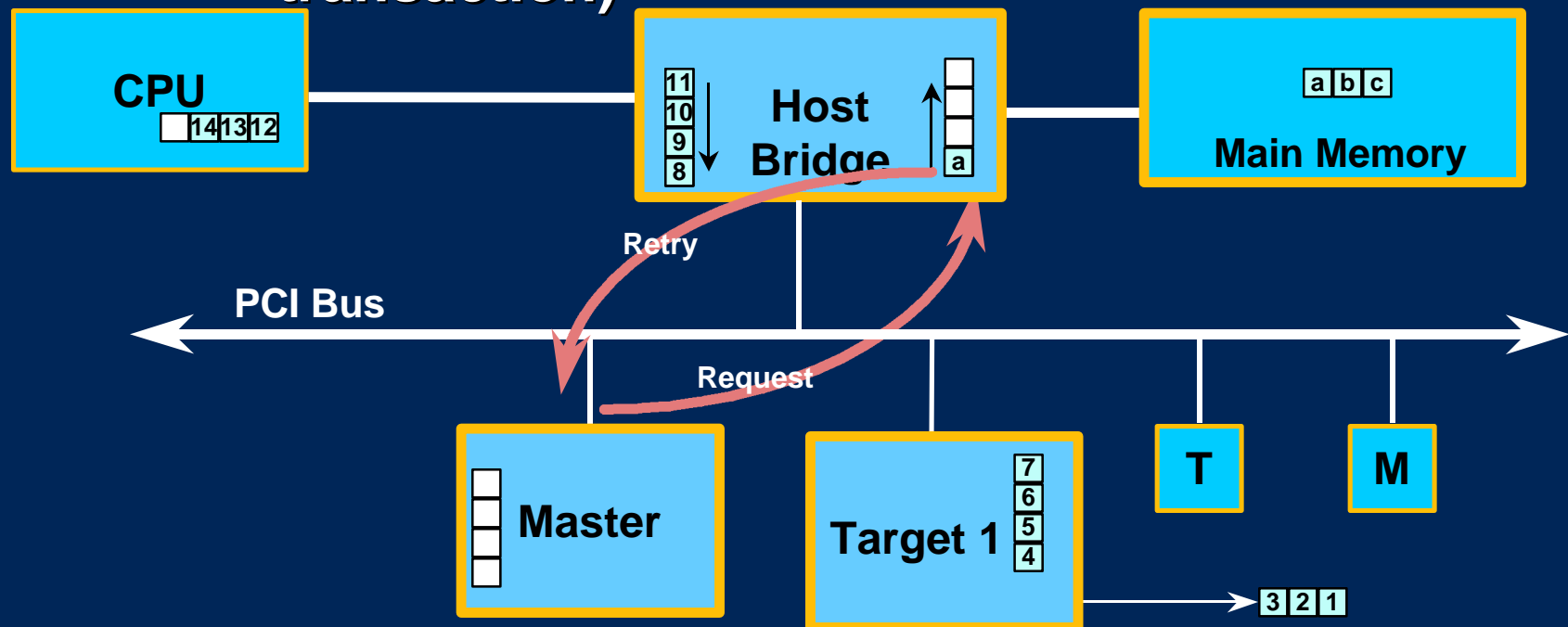
Blocked Access

- ◆ CPU is writing data to Target 1
 - ◆ CPU writes fast than target can accept
 - ◆ Target 1 buffers some
 - ◆ Host Bridge Buffers some more



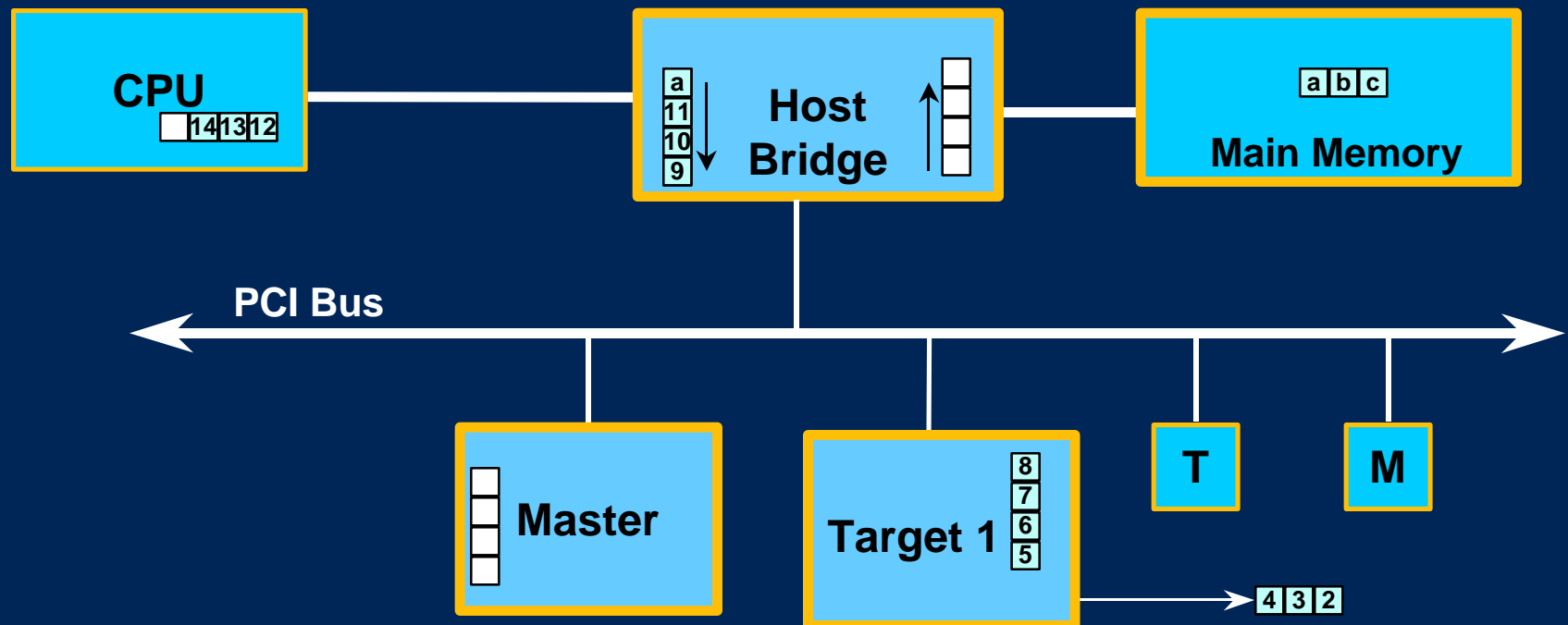
Blocked Access

- ◆ Master tries to read data from main memory
- ◆ Host Bridge required to flush posted write data before returning read data
 - ◆ Retry (no delayed transaction)
 - ◆ Retry (enqueue request - delayed transaction)



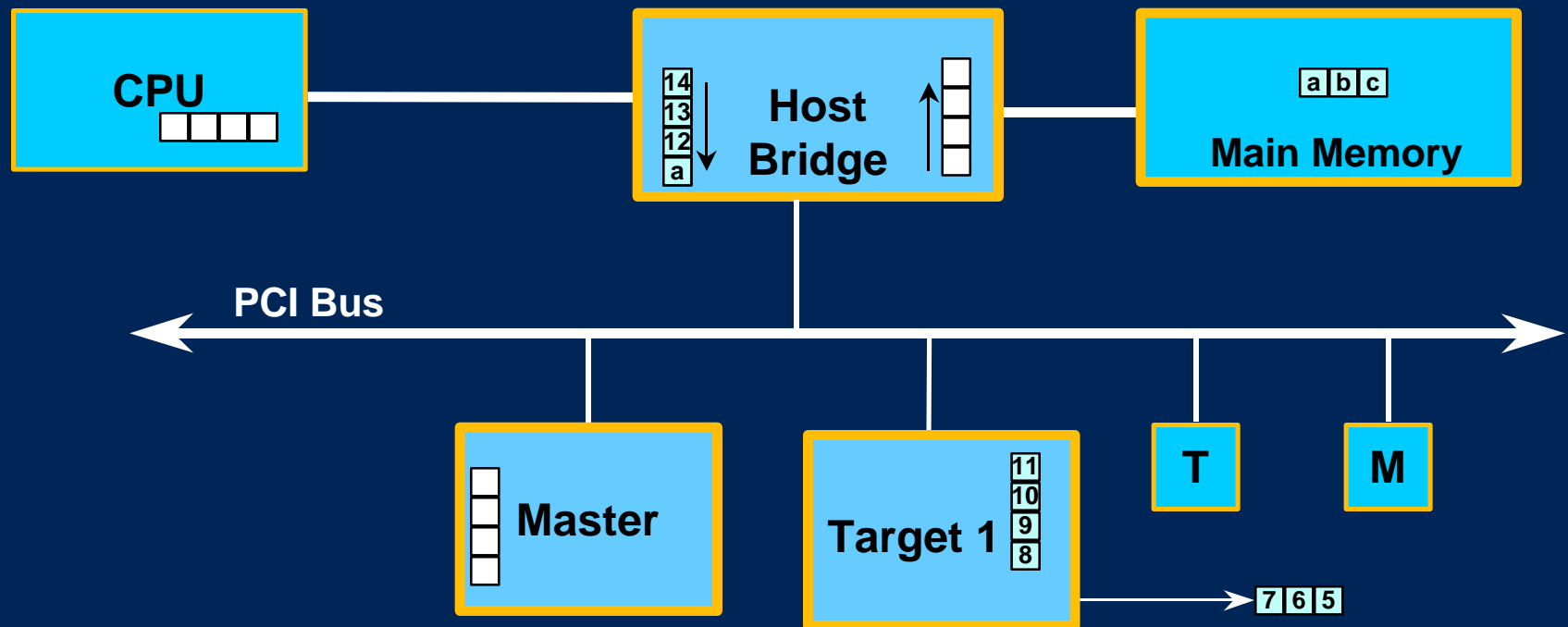
Blocked Access

- ◆ Host Bridge continues to attempt to flush buffers
- ◆ If Host Bridge does Delayed Transaction then empty slot can be used to move read data



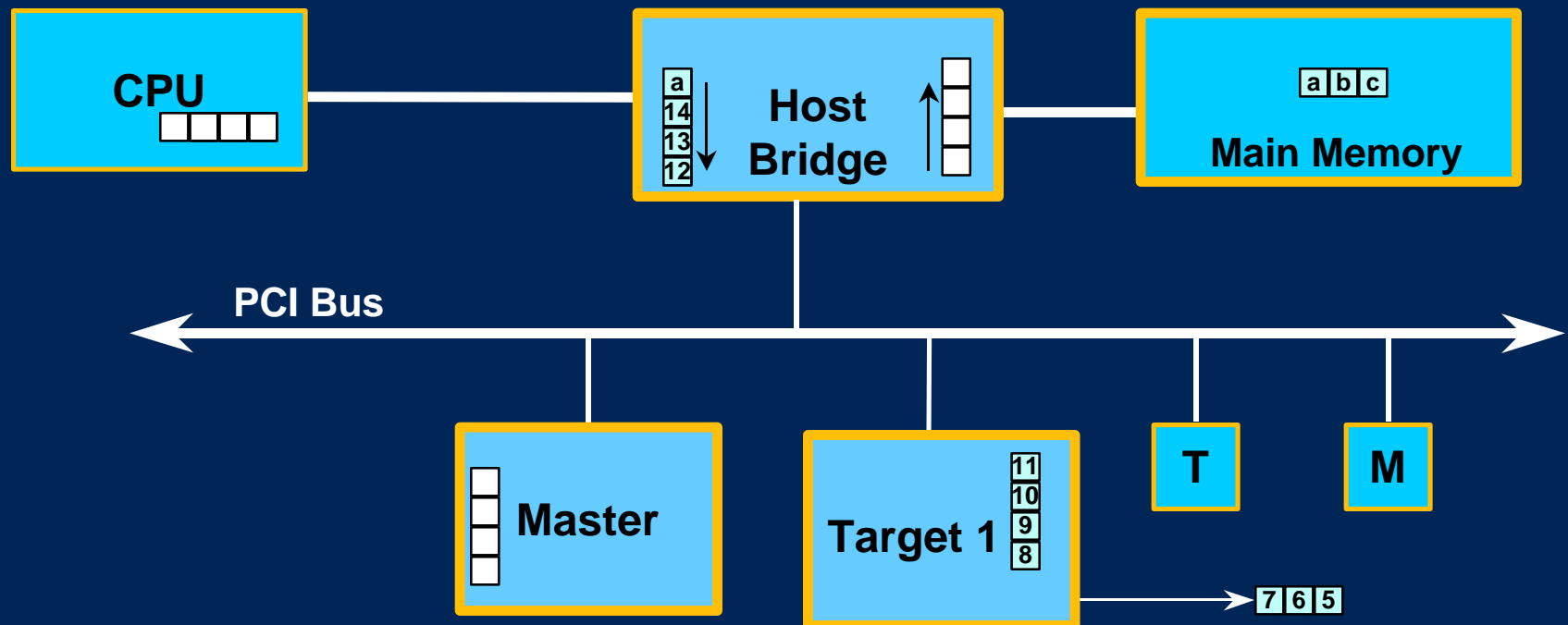
Blocked Access

- ◆ Eventually the master's read succeeds
- ◆ The elapsed time from the initial read to the data transfer depends on how fast the target consumes data



Blocked Access

- ◆ When Host Bridge does not do Delayed Transaction, the read is delayed until CPU completes all write operations



How we address the problem

- ◆ Make sure drivers ship in 'good citizen' mode
- ◆ Intel is driving PCI spec change to restrict Memory Write retry time
 - ◆ sets 10 microsecond limit
- ◆ Specialized add-in card flags violations
 - ◆ deployed at PCI Compliance Workshops
 - ◆ Available through PCI SIG web pages

String Descriptors

String Descriptors

- ◆ If your device doesn't have any, make sure all string indexes are zero
- ◆ If supported, string index zero returns a list of supported language IDs
 - ◆ String index zero is a 'string descriptor'
 - ◆ Type is string, must have length byte

04h	03h	09h	04h
-----	-----	-----	-----

Size

Type

Language ID
(English in this case)

*Unicode values
(defined in NT DDK)*

String Descriptor Usage

- ◆ Helps identify a device during enumeration
- ◆ iProduct string is used instead of “Unknown Device”



String Descriptor Usage

- ◆ iSerialNumber is used to uniquely identify device in registry
- ◆ Allows 'silent' PNP (e.g.; no UI) for all installs after the first one
- ◆ Use of serial numbers is encouraged

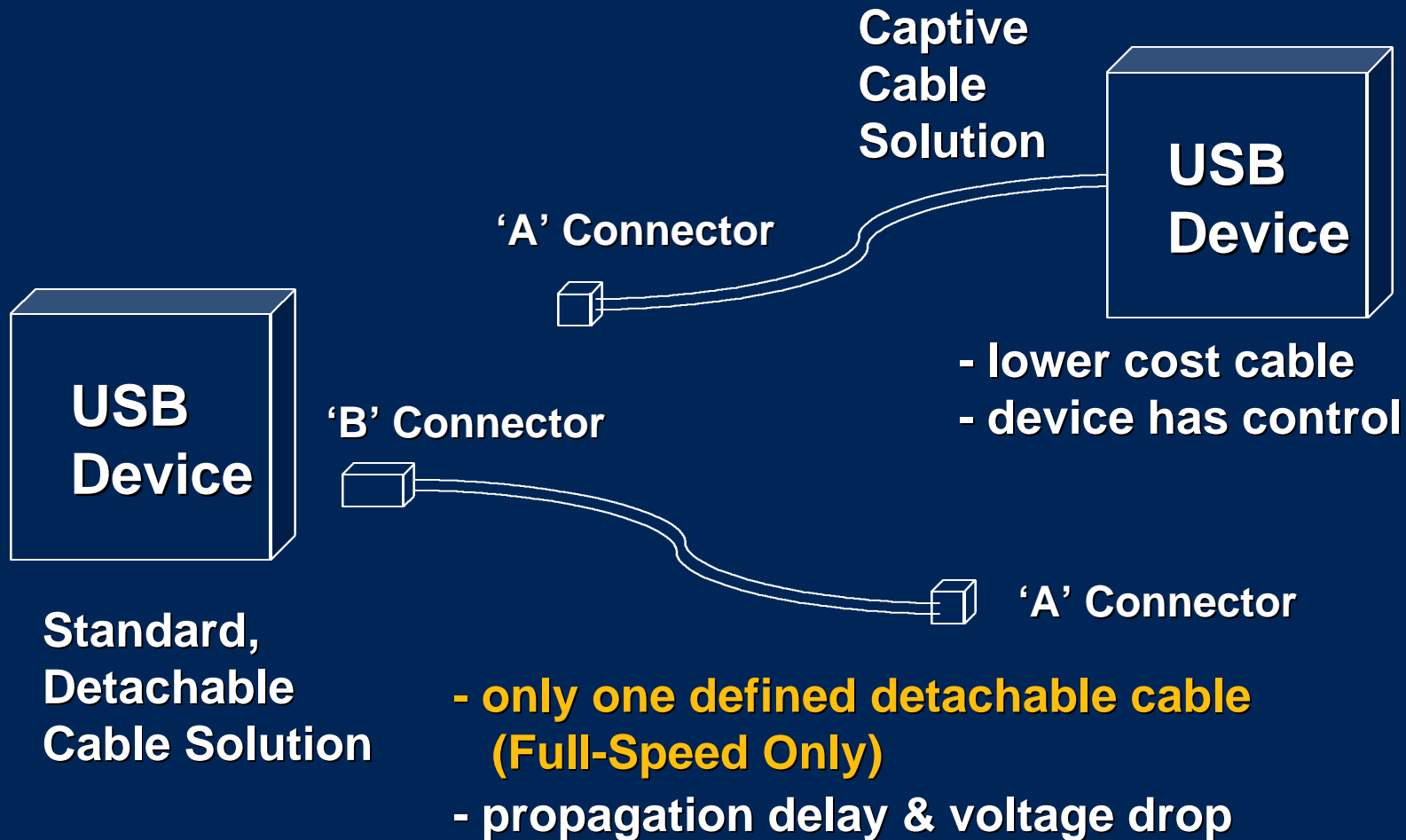
String Descriptor Usage

- ◆ Other string descriptors may be available through class interfaces
 - ◆ HID allows access from applications
 - ◆ Peripherals, systems, silicon, hubs, cables

USB Cables/Connectors

USB Cables/Connectors:

Example: *USB Cabling Options*



USB Cables/Connectors

- ◆ **Extension Cables Prohibited in 1.1**
 - ◆ Any cable with Series A or Series B socket
 - ◆ Watch out for illegal monitors
 - ◆ provide single USB port and don't contain a hub

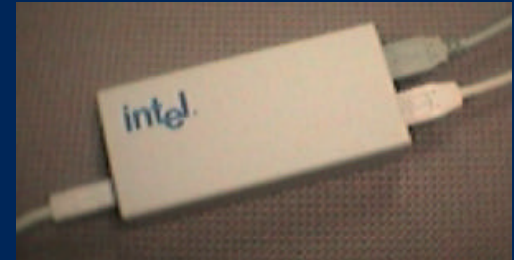
Why no Extension Cables?

- ◆ USB Spec very explicit about cable propagation delays and socket minimum voltages
- ◆ Devices with attached cables minimize cost by tight design to spec

**Extension cable will cause
these devices to fail**

What's a user to do??

- ◆ USB hubs are the answer
- ◆ Hubs provide an effective and approved way to 'extend' a USB connection



**Inrush Current is not
equal to Overcurrent**

Overcurrent issues

Device Behavior

Inrush current at connect
and configuration is
expected and legal

Safety Issues

Conform to
applicable regulations

**Some current limiters are
built to be current 'cops'**

**These are not good. Current
limiting should be for safety only.**



Call to Action

- ◆ **Manage your bandwidth, or it will manage you**
- ◆ **Power management is important; learn to optimize USB contribution to power savings and instantly available PC**
- ◆ **Study the USB Core Specification R1.1**
 - ◆ **Take advantage of clarifications**
 - ◆ **Interrupt Out**
 - ◆ **Higher suspend current**

Call to Action

- ◆ **Be aware of system interactions**
 - ◆ **Bus powered hubs can cause topology restrictions**
 - ◆ **PCI 'bad citizens' may effect USB**
 - ◆ **Use string descriptors**
 - ◆ **Don't use extension cables**
 - ◆ **Beware of current limiters**